IN THE ABSTRACT:

Please replace the abstract by the attached abstract on a separate sheet.

REMARKS

Claims 1-4 and 6-21 are pending. Claims 1 and 7 have been amended. New claim 14-21 have been added. No new matter has been introduced. Applicant believes the claims comply with 35 U.S.C. § 112.

Claims 1-4 and 6-13 stand rejected under 35 U.S.C. § 102(a) as being anticipated by Cho et al. Claims 1-4 and 6-13 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Ray et al.

Applicant respectfully submits that independent claim 1 is novel and patentable over Cho et al. and Ray et al. because, for instance, they do not disclose or suggest providing a first instruction causing loading of a first part of the unaligned data into a first storage location by using a first pointer giving a memory address of a first position, and rotating and masking and sign-extending the first part of the unaligned data in the first storage location from the first position to a second position.

As described in the specification, for example, at page 7, lines 1-19, a sign-extension is performed on the first storage location when the first part of the unaligned data is in the second position of the first storage location. Because computer architectures go from 32 bits and 64 bits and even 128 bits, it is desirable to provide a better method of handling unaligned data, which includes proper sign extension as recited in claim 1.

Neither Cho et al. nor Ray et al. discloses the sign extension as recited in claim 1. Cho et al. merely discloses an aligner that aligns data for load-store instructions and shifts or rotates data for arithmetic logic instructions. Ray et al. is directed to a superscalar processor which allows speculative execution of misaligned cache access instruction. These references are devoid of any disclosure or suggestion of using sign extension to better handle unaligned data as claimed.

For at least the foregoing reasons, claim 1 and claims 2-4, 6, and 13 depending therefrom are novel and patentable over the cited references.

Applicant respectfully submits that independent claim 7 is novel and patentable over Cho et al. and Ray et al. because, for instance, neither of them discloses or suggests providing a first instruction causing rotation of data in a first storage location and sign-extending and storing of a first part of the data in a first portion of unaligned plurality of memory locations from a first position to a second position. As discussed above, Cho et al. and Ray et al. fail to teach or suggest using sign extension to better handle unaligned data as claimed.

For at least the foregoing reasons, claim 7 and claims 8-12 depending therefrom are novel and patentable over the cited references.

Applicant respectfully asserts that new independent claim 14 is novel and patentable over Cho et al. and Ray et al. because, for instance, they do not teach or suggest loading a first part of the unaligned data into a first storage location, and rotating and sign-extending the first part of the unaligned data in the first storage location from a first position to a second position.

For at least the foregoing reasons, claim 14 and claims 15-18 depending therefrom are novel and patentable over the cited references.

Applicant respectfully asserts that new independent claim 19 is novel and patentable over Cho et al. and Ray et al. because, for instance, they do not teach or suggest rotating and sign-extending a first part of the data in a first storage location from a first position to a second position, and storing the data located in second position in the unaligned plurality of memory locations at an address given by a first pointer.

For at least the foregoing reasons, claim 19 and claims 20-21 depending therefrom are novel and patentable over the cited references.

CONCLUSION

In view of the foregoing, Applicant believes all claims now pending in this Application are in condition for allowance and an action to that end is urged. If the

Examiner believes a telephone conference would aid in the prosecution of this case in any way, please call the undersigned at 650-326-2400.

Respectfully submitted,

Chun-Pok Leung
Reg. No. 41,405

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 650-326-2400
Fax: 415-576-0300
RL:rl
PA 3302026 v1

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE CLAIMS:

Please amend claims 1 and 7; and add new claim 14-21 as follows.

1. (Twice amended) A method for loading unaligned data stored in a plurality of memory locations, comprising:

providing a first instruction causing loading of a first part of said unaligned data into a first storage location by using a first pointer giving a memory address of a first position;

rotating and masking and sign-extending said first part of said unaligned data in said first storage location from **[a]** the first position to a second position;

providing a second instruction causing loading of a second part of said unaligned data into a second storage location by using a second pointer giving a memory address of a fourth position;

rotating and masking said second part of said unaligned data in said second storage location from a third position to **[a]** the fourth position; and

providing a third instruction causing combining of said first storage location with said second location using a logical operation into a result storage location.

7. (Twice amended) A method for storing data into an unaligned plurality of memory locations, comprising:

providing a first instruction causing rotation of data in a first storage location and sign-extending and storing of a first part of said data in a first portion of unaligned plurality of memory locations from a first position to a second position;

having a first pointer giving an address of **[a]** the first position;

providing a second instruction causing rotation of data in a second storage location and storing of a second part of said data in a second portion of unaligned plurality of memory locations from a third position to a fourth position; and

having a second pointer giving an address of **[a]** the fourth position.

--14. A method for loading unaligned data stored in a plurality of memory locations, comprising:

loading a first part of said unaligned data into a first storage location;

rotating and sign-extending said first part of said unaligned data in said first storage location from a first position to a second position;

loading a second part of said unaligned data into a second storage location;

rotating said second part of said unaligned data in said second storage location from a third position to a fourth position; and

combining said first storage location with said second location using a logical operation into a result storage location.

15. The method of claim 14 wherein said first storage location is a first register, said second storage location is a second register, and said result storage location is a result register.

16. The method of claim 15 wherein said registers are 64-bits in length.

17. The method of claim 14 wherein the logical operation is a bit-wise OR operation.

18. The method of claim 14 wherein said rotating is performed in two phases comprising a first phase in which a major rotation is performed and a second phase in which a minor rotation is performed.

19. A method for storing data into an unaligned plurality of memory locations, comprising:

rotating and sign-extending a first part of said data in a first storage location from a first position to a second position;

storing said data located in second position in said unaligned plurality of memory locations at an address given by a first pointer;

rotating a second part of said data in a second storage location from a third position to a fourth position; and

storing said data located in forth position in said unaligned plurality of memory locations at an address given by a second pointer.

20. The method of claim 19 wherein said first storage location is a first 64-bit register, said second storage location is a 64-bit second register, and said result storage location is a 64-bit result register.

21. The method of claim 19 wherein said data is selected from a group consisting of data 16, 32, and 64 bits in length.--